NPS-CS-13-001

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

EXAMINING THE EFFECT OF ORGANIZATIONAL ROLES IN SHAPING NETWORK TRAFFIC ACTIVITY

by

Jeffrey Dean, Geoffrey G. Xie, Neil Rowe, and Robert Beverly

August 2012

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From-To)*
29-08-2012	Technical Report	

4. TITLE AND SUBTITLE
Examining the Effect of Organizational Roles in Shaping Network Traffic Activity

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Jeffrey Dean, Geoffrey G. Xie, Neil Rowe, and Robert Beverly

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) AND ADDRESS(ES)
Computer Science Department
Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER
NPS-CS-13-001

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

We hypothesize that a computer user's environment shapes the characteristics of his/her network traffic. In particular, we focus on whether a user's role at the work place induces discriminating characteristics, due to the task requirements of assuming that role. If true, this shaping can enable development of useful indicators for detecting insider threat activities.

We develop a methodology to evaluate this hypothesis, characterized by (i) new traffic similarity metrics for quantifying the variations of flow-level traffic activities between role-based user groups; (ii) use of exclusively Netflow data to build user/group discriminating features; and (iii) a rigorous process for attributing flows to users and mapping users to roles.

We evaluate the role-based hypothesis using a four-week long dataset of Netflow records from a university building. We measure inter-system similarities using several flow based methodologies, and show significant levels of value overlap when computing inter and intra role-based group similarities. We did observe indications that similar roles lead to similar allocations of time for related tasks. We also found that most of the user traffic features under consideration persist over time, with a typical similarity value of above 0.8 week to week. These findings lead us to believe that measuring role based group characteristics on the network requires a temporal component for the characterization to be useful.

15. SUBJECT TERMS
Netflow, Behavior Detection

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Jeffrey Dean
Unclassified	Unclassified	Unclassified	UU	40	**19b. TELEPHONE NUMBER** *(include area code)* 831-656-3696

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

THIS PAGE INTENTIONALLY LEFT BLANK

NAVAL POSTGRADUATE SCHOOL
Monterey, California 93943-5000

RDML Jan E. Tighe
Interim President

O. Douglas Moses
Acting Provost

The report entitled *"Examining the Effect of Organizational Roles in Shaping Network Traffic Activity"* was prepared for the Computer Science Department of the Naval Postgraduate School.

Further distribution of all or part of this report is authorized.

This report was prepared by:

Jeffrey S. Dean
PhD Candidate
Computer Science

Geoffrey G. Xie
Professor
Computer Science

Neil Rowe
Professor
Computer Science

Robert Beverly
Assistant Professor
Computer Science

Reviewed by:

Peter J. Denning
Chairman, Computer Science

Released by:

Jeffrey D. Paduan
Vice President and
Dean of Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

We hypothesize that a computer user's environment shapes the characteristics of his/her network traffic. In particular, we focus on whether a user's role at the work place induces discriminating characteristics, due to the task requirements of assuming that role. If true, this shaping can enable development of useful indicators for detecting insider threat activities.

We develop a methodology to evaluate this hypothesis, characterized by (i) new traffic similarity metrics for quantifying the variations of flow-level traffic activities between role-based user groups; (ii) use of exclusively Netflow data to build user/group discriminating features; and (iii) a rigorous process for attributing flows to users and mapping users to roles.

We evaluate the role-based hypothesis using a four-week long dataset of Netflow records from a university building. We measure inter-system similarities using several flow based methodologies, and show significant levels of value overlap when computing inter and intra role-based group similarities. We did observe indications that similar roles lead to similar allocations of time for related tasks. We also found that most of the user traffic features under consideration persist over time, with a typical similarity value of above 0.8 week to week. These findings lead us to believe that measuring role based group characteristics on the network requires a temporal component for the characterization to be useful.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

I. I. INTRODUCTION

Monitoring and profiling network traffic is an essential function of network management, used to protect systems from threats both external and internal. To accommodate diverse and ever-changing threats, anomaly-based approaches are frequently used to supplement misuse intrusion detection systems. Anomaly-based systems form a notion of "normal" traffic and alert on any deviations from this norm [1]. Unfortunately, determining what constitutes normal user, host, or application behavior can be difficult in practice.
In this paper, we take inspiration from prior work in Role Based Access Control (RBAC) [2] and explore the impact of organizational roles on observable network traffic activity. Further, we examine the extent to which roles can be leveraged to better generalize per-role legitimate traffic characteristics.

In particular, we test the hypothesis that membership in role-related groups (e.g. clerical, engineering, students, etc.) bounds user network activity patterns, and that this bounding will be detectable in the distributions of features derived from Netflow records. Intuitively, one might expect a group of users with similar organizational roles to exhibit spatial and temporal locality, running the same set of applications, accessing similar network resources, etc. To better understand the impact of roles in shaping traffic, we measure several similarity metrics between users of the same role-based group, as well as between users of different roles.

Complicating the validation of this hypothesis however are other non-human, non-group factors that shape observable network traffic feature distributions, or obscure their effects. For example, variations in system software, operating systems and systems hardware can contribute to traffic diversity over and above the variations caused by system users. While this may not be as big a problem among rigid IT deployments, many companies and educational institutions allow greater flexibility in system configurations. These variations may add noise to the problem. We explore the relationship between operating systems and a set of network traffic features used for our analysis, and compare this with the relationship between user roles and the features.

We use live traffic from a campus building network to test our hypotheses. Importantly, we use external information (e.g. mapping between MAC and IP addresses) to attribute captured traffic flows to known users and determine each user's organizational role (e.g. student, faculty, admin., etc.). The additional processing provides for a rich dataset where flows are labeled with ground-truth.

In this work, we focus on behavior that may be inferred from NetFlow records. NetFlow is an attractive target due to its ubiquity. We categorize 35 known user systems into five role-based groups. Using 34 features derived from these users' NetFlow records over the course of one month, we apply a variety of machine learning techniques to understand the impact of roles on traffic activity. Our primary contributions include:

The development of a methodology for examining the effect of organizational

roles in shaping network traffic activities. The methodology is characterized by three new traffic similarity metrics for quantifying the variations of flow-level traffic activities between user groups, the use of exclusively Netflow data to build user/group discriminating features, and a rigorous process for attributing flows to users and mapping users to roles.

To the best of our knowledge, this paper is the first in evaluating the role effect solely based on traffic features derived from Netflow data. The results reveal some evidence of clustering of traffic features around roles. In addition, the clustering appears to reflect similar allocations of time to related tasks. We also show that most of the user traffic characteristics considered in this paper seem to persist over time, with a typical similarity value of around 0.8 week to week.

The rest of the paper is organized as follows. Section 2 provides a brief overview of related work, followed by a detailed description of our methodology in Section 3. We present the evaluation results in Section 4, and discuss possible interpretations of the results as well as areas for future work in Section 5. Section 6 concludes the paper.

II. II. RELATED WORK

Developing user profiles based on group behaviors is not a new concept. In Dorothy Denning's landmark paper [1], she discussed the utility of profiling users or groups of users in order to detect deviations in behavior. Denning noted that "Aggregate individual activity reveals whether the behavior of a given user (or object) is consistent with that of other users (or objects)." Anderson et al. demonstrated this concept using IBM's Identity Risk and Investigation Solution (IRIS) system [3]. Measuring features like the number of accesses to an application or login time of day they created a set of user profiles, and applied the concept of peer groups to develop expected norms of behavior. While this approach shows promise in leveraging group norms for monitoring behavior, the features used require access to application layer network data (or a monitoring application on each system), a much more intrusive level of monitoring than use of Netflow based features.

Park and Giordano leveraged Role Based Access Control (RBAC) concepts [4] as a means of narrowing acceptable behavioral thresholds. Thresholds based on behavior related metrics were set by using the ranges observed among role-based groups. While the features used were not Netflow based, this paper did emphasize use of user roles (or groups) and historical behaviors as means for setting normal behavioral limits. Another study on the use of RBAC principles [5] used activity logs to compare behaviors, and found that the use of roles improved accuracy for detecting malicious insiders.

Frias-Martinez [6] examined the use of system/user network behaviors as a means of applying Behavior-Based Network Access Control (BB NAC). Without employing external labels such as user roles, the BB NAC controller adds a new system to a network group based on similarities of behavioral profiles. These profiles are based on each system's per-port network statistics, and similarities are computed by clustering these statistics to establish inter-system distances. Once in a group the new system's network usage is monitored, and the group can "vote" on whether the system still fits in with the cluster norm. This approach is based on the assumptions/observations that computer usage statistics tend to be stable over time, and that usage behaviors tend to "cluster" in separable groups.

The use of Netflow data is a common practice in traffic analysis and anomaly detection. Recent work showed that it is possible to mine Netflow data to detect worms and server-like behaviors [7], and identify some applications [8] [9]. Karagiannis et al. used graphlets [10] based on Netflow level features to profile host applications usage. Nodes in a graphlet capture the kinds and number of connections created during some set period of time, embodied in links between protocol, destination IP, source port, destination port nodes. Profile graphlets are trimmed down to contain only "significant nodes", which were defined as those with in-degree and out-degree counts greater than one. Using graphlets, it was possible to recognize usage of a number of applications by the user. Netflow level data has also been used to recognize changes in user behavior that were dependent on user working locations [11].

III. III. METHODOLOGY

At a high level, our research methodology involves:

Extracting features from one month's worth of ingress and egress NetFlow traffic records from a university academic building.

Leveraging external ground-truth to map IP sources to users, and users to organizational roles.

Applying traffic similarity metrics and other analysis techniques to quantify how distinct the traffic features of users (or groups of users) are.

This section details each of these steps, highlighting major challenges and approaches.

A. DATASET

We collected NetFlow records over a one-month period (July 18 to August 14, 2011) from a campus large academic building consisting of four academic departments, hundreds of people, as well as dozens of classrooms and computer labs. All non-lab end systems attached to the building's wired infrastructure were under a single /21 subnet. A total of 892 unique IP addresses were present in the traffic trace.

To get ground truth on a set of computer users on the network and their group affiliations, we solicited volunteers from each of the academic departments within the building in January, 2012. In that survey, we asked for information about roles, software (including the operating system) and computer IP and MAC address(es). We received 53 responses to the survey. System configurations varied: 41 were Windows 7 or XP, 22 were MacOS, and seven were Linux distributions. As this was an academic environment many users commonly made use of virtual machines, running other operating systems along with the host OS. These were not detailed in the survey results. For our analysis, operating system descriptions were not specified down to a specific Windows service pack level, MAC OS version or Linux distribution.

Using the system MAC addresses, we isolated the packets passed to/from systems belonging to each volunteer from the data set prior to converting the packet data into Netflow records. Extracting packets from capture files based on the MAC address enabled tracking user activities even IP addresses are reassigned by the DHCP server.

Unfortunately, not all volunteer data was usable for analysis. Some volunteer hosts were either totally absent or sparsely active over the collection period. Some volunteers had changed systems between the collection period and the time of the survey. For these reasons, only 34 of the volunteer systems remained as viable study subjects.
The roles we selected were somewhat broad in scope: PhD student, Administrative, Research Associate, Lecturer and Professor. While broad however, these categories represent sets of responsibilities and tasks with limited overlap, much like the ”multiple hat” roles found in most organizations. In addition, given a sampling set of 34 volunteers, creating more specific roles would have created much smaller groups.

B. IDENTIFYING OTHER USER FLOWS

While we had isolated Netflow data for the systems used by our volunteers, we also felt it was necessary to identify a pool of non-attributed user systems as a reference on normal user traffic. This required scrubbing our collected data to eliminate any servers within the address space.
To this end, we:

- Dropped addresses providing services on SMTP, print server protocol, DNS, HTTP, POP3, NNTP, IMAP, SNMP, Service Locator Protocol, and HTTPS related ports.
- Eliminated systems using fewer local ports than distant ports (typical of server behavior)
- Cut systems with least twice the data going out as coming in during the test period
- Dropped those systems with no apparent HTTP traffic, as use of the web has become such an essential human behavior.

Role	Group Size	Flow Count
Professor	15	5,435,523
Lecturer	6	1,824,184
Research Associate	5	1,280,757
Admin	3	548,817
PhD Student	5	1,807,773
Other	399	103,630,469

Table 1: User groups considered by this study

We also dropped those IP addresses that had been active 10 days or less, in order to have enough data samples to compare in a meaningful manner. This address culling process reduced the remaining IP address pool to 399 distinct addresses, which we refer to collectively as the "Other" user group. While this culling process certainly removed a number of systems that were not servers, it did provide us with a pool of systems for which we had high confidence were not automated. This pool was our "control group," in that we drew random groups of systems from the pool to test the null hypothesis, i.e. that belonging to a role based group does not impact the feature values we derived from Netflow. Table 1 summarizes the user groups extracted from the data set.

C. EXTRACTING FEATURES

To understand the influence of roles on network behavior, we focused on those flows generated during working hours, 0800–18:00 M-F (local time relative to collection). This was done to capture most common workday activities, as a means of investigating group network behaviors. As malicious users are known to not restrict their activities to business hours, we plan to expand this window at a later point to evaluate anomalous off hours activities.

The captured flows for each user were divided into sample sets of 15 minute intervals. Flows during each interval were analyzed, and features extracted based on the network activity observed. The interval of 15 minutes was chosen to provide enough flows to generate statistically meaningful features, and to capture user generated traffic representing from one to a small number of individual tasks.

The maximum possible traffic from each user consisted of a total of 10 (hours per day) × 4 (samples per hour) × 5 (days per week) × 4 (weeks) = 800 sample flow sets. Flows spanning more than one sample period were split accordingly, so that volumetric data (byte counts, etc.) would be represented during the correct period.

1	port53Bytes	Total bytes sent to port 53 (DNS)
2	port80Bytes	Total bytes sent to port 80 (HTTP)
3	port443Bytes	Total bytes sent to port 443 (HTTPS)
4	port993Bytes	Total bytes sent to port 993 (IMAP)
5	numFlows	Number of flows per interval
6	flagMetric	Entropy measure of flag use among flows (SYN)
7	direction	Fraction of flows that are outgoing
8	aveIntPktTime	Standard deviation of average inter-packet time
9	packets	Total outgoing flow packets
10	bytes	Total outgoing flow bytes
11	localPorts	Total distinct local ports in flows
12	distPorts	Total distinct distant ports in flows
13	interFlowTime	Standard deviation of inter-flow arrival times
14	duration	Standard deviation of flow duration times
15	numDist	Number of distant IP addresses per interval
16	protocol	Fraction of flows using a given protocol (TCP)
17	bytesPerPacket	Average of flow bytes per packet
18	emptyTCP	Entropy of TCP flow counts with no payload data
19	serviceNets	Entropy of distant IP address/ports
20	ipDistance	Std deviation of IP distances of all flows
21	portsPerFlow	Average number of distinct distant ports
22	addrPerFlow	Average number of distinct distant IP addresses
23	addrDist	Std deviation of IP distances of all flows
24	portDist	Std deviation of local/distant port numbers of all flows
25	notTCPUDP	Graphlet: # out-degree links from other proto. nodes
26	TCP	Graphlet: # out-degree links from TCP nodes
27	UDP	Graphlet: # out-degree links from UDP nodes
28	maxDistIP	Graphlet: max out-degree count for distant IP node
29	maxLocPort	Graphlet: max out-degree for a local port node
30	maxLocPortIn	Graphlet: max in-degree for a local port node
31	maxDistPort	Graphlet: max out-degree for a distant port node
32	maxDistPortIn	Graphlet: max in-degree for a distant port node
33	maxDistAddrIn	Graphlet: max in-degree for a distant IP addr node
34	serverRatio	Graphlet: ratio of maxLocPort to maxDistPort

Table 2: Features evaluated

We extracted 34 features from each flow set, as described in Table 2. Some features were intuitive choices, e.g., the amount of traffic to/from well-known server ports or the standard deviation of inter-flow arrival times. For such features, we built in the capability of filtering the flows based on flow direction, local or distant port numbers, or the protocol used. Flow derived numerical values could be summed, or have the mean or standard deviation computed. For categorical features (flags, ports, protocols or IP addresses), the flow derived values can be counted (e.g. total number different ports) or the normalized entropy computed. Thus even a limited set of basic features could be used to produce a diverse set of values describing an interval's flows. Other features were inspired from the literature, e.g., the count of flows with empty or non-empty payload [8] for a given protocol, computing a distance between local and distant ports and addresses [12], and the graphlet concept developed in the BLINC work [10].

The graphlet features were the most complex to define and extract. A graphlet is a directed graph with each path, from a common starting node, encoding a unique five tuple <source IP, protocol, distant IP, source port, distant port> found in a set of flows. A graphlet can succinctly represent the level and types of diversity found in the flow data.

To try and capture some of the information embedded in the structure of a graphlet, we created features based on the number of in-degree and out-degree connections of the nodes. For each interval, the maximum in-degree and out-degree connection counts for the local port, distant port, and distant IP address nodes were captured. We also extracted the number of out- degree connections for the TCP and UDP protocol nodes, and from a catch-all (notTCPUDP) protocol node. As servers tend to have flows using a few local service ports and many distant ports, another feature based on the ratio of the maxLocalPort to maxDistPort was created. While these values fail to capture all the semantic meaning within a graphlet structure, they do provide some insight.

Two of the features we tested, serviceNets and ipDistance, were created for these experiments. The former measures the entropy of the distant addresses and ports visited, as a measure of the variability of sites/services visited by the user. The entropy was computed by merging the distant IP address and port values into a single pattern (e.g. 157.166.226.25 port 80 becomes 157.166.226:80). Commonly, multiple server hosts in one /24 subnet can provide the same service (e.g., CNN); this merging of address and port information helps consolidate sites providing a common service.

A simple IP distance metric was used to represent the difference between a flow's source and destination IP addresses, formally ipDistance = log(|srcIP − destIP |), where srcIP and destIP are interpreted as 32-bit integers. ipDistance is usually captured as the standard deviation of the local/distant IP address distances for all flows in the interval, providing a slightly different view on destination address diversity. Other IP address distance metrics exist. A more realistic IP address measure (addrDist) was derived from Coull et al. [12], in which IP addresses are compared based on category (Unicast: public, private; Other: multicast, broadcast, link local, default network). This alternate IP distance metric was not tested at this time.

D. NOTATION

Comparisons for this analysis were based on feature vectors, using the 34 features described above. Each feature vector consisted of 34 feature values, representing the network activity of a system between times *t* and Δt (Δt = 15 minutes). Each system was associated with one user (who may operate one or more systems), and each user was associated with one role. These relationships were in part an effect of the user/system selection process applied for these tests; in many networks systems may be used by more than one user and a user may have more than one defined role.

Thus for users $U = \{u_1, u_2, \dots, u_m\}$, systems $S = \{s_1, s_2, \dots, s_n\}$ and roles $R = \{r_1, r_2, \dots, r_p\}$, we can represent a feature vector A tied to user *i* associated with role *j* using system *k* at time *t* as $A = F[i][j][k][t]$.

For equations in which specific elements within a vector are referenced, say the *i*th component in vector A, the value is denoted as A_i.

E. FEATURE CORRELATIONS WITH ROLES/OS

We correlated user membership in a specific role (indicated with a binary vector) with the features in each feature vector, using Pearson's algorithm. Each feature vector $F[i][j][k][t]$ was prepended by a p length binary vector (for the p roles, membership/ non-membership denoted by 1/0), creating a new vector $F'[i][j][k][t]$. For 34 features plus 5 roles, this created a 39 x 39 correlation matrix.

For comparison purposes we also correlated each feature vector to each type of OS found in the dataset, where $OS \in \{Linux, Mac, Windows\ XP, Windows\ 7\}$, using the same approach as used for role correlation.

F. SIMILARITY MEASURES

Ideally, our features should enable some measure of how similar (or dissimilar) different computer users or groups are in terms of computer usage. One potential approach to similarity measurement would be to compare the distributions of feature values between users and/or groups. If usage behavior is consistent relative to a given feature, this should be reflected in recurring value distributions within groups.

1. Feature Distributions:

To compare systems based on feature distributions, we expressed the distributions as histograms (i.e. Probability Mass Function – PMF). Each histogram bin covers a subset of the total feature value range. Bin value ranges could be linearly spaced, but for features with large values (such as total byte counts) the bin ranges were spaced non-linearly (exponential growth). Using non-linear spacing can address the "mice and elephants" nature of network traffic by spreading out smaller value ranges and compressing larger ones. Additionally, as we denoted periods of no traffic using zero

values for each feature, we found that creating a special "zero bin" was useful to represent periods of inactivity.

Our initial attempt at evaluating the similarities of feature distribution vectors was to apply a simple cosine similarity computation:

$$S_{cosine} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2}\ \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

While algorithmically simple, this approach turned out to be the least useful. The resulting similarity values obtained when comparing two groups varied widely, to the extent that group to group similarity plots (such as in Figures 2-4) showed the 95% confidence intervals occupying much of the [0,1] range for most features.

As a means of exploring this approach to comparing users and groups, we evaluated two other methods for comparing feature value distributions as a measure of similarity. These were termed the Bin Ratio measure and the Kolmogorov–Smirnov (KS) based measure. This last measure was derived from the Kolmogorov–Smirnov test for Empirical Distribution Function (EDF) equivalence. Each measure returned values between [0,1].

a. Bin Ratio Measure:

To focus on the proportionality of distribution values on a bin by bin basis, we created the bin ratio measure. This measure is determined by computing the value ratio for each corresponding pair of PMF distribution bins, and using the average ratio value across the distribution as the similarity measure. Formally:

$$S_{ratio} = \frac{1}{n}\sum_{i=1}^{n} \min\left(f_i^A, f_i^B\right) / \max\left(f_i^A, f_i^B\right)$$

where f^A and f^B are PMF distributions of feature f from systems A and B.

b. Kolmogorov–Smirnov (KS) Based Measure:

The Kolmogorov–Smirnov test for two distributions essentially involves overlaying the EDFs of the two distributions and computing the maximum vertical distance between the two curves.

For our variation of the KS test as a measure of similarity, we convert the PMFs of a given feature for two systems into EDFs. From these we compute the ratio between the "difference area" (Figure 1) and the total area under the two EDF curves. We subtract this ratio from one; if the EDFs are identical there is no area between the curves and the KS similarity measure is 1.0.

Formally, the KS similarity metric is defined:

$$S_{KS} = \frac{1}{n}\sum_{i=1}^{n} \min\left(F_i^A, F_i^B\right) / \max\left(F_i^A, F_i^B\right)$$

where F^A and F^B are feature EDF distributions from systems A and B.

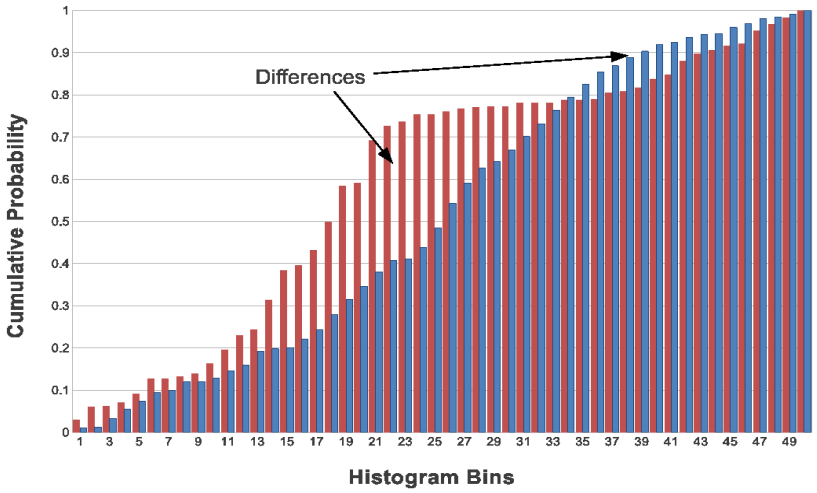

Figure 1: Kolmogorov-Smirnov Based Measure

G. MACHINE LEARNING

For testing the separability of the groups based on the feature vectors, we applied machine learning tools (a J48 decision tree from Weka [14], the Relief algorithm from Orange [15], and a k-means clustering program). The ML techniques were applied in different ways, each providing a different view of the data.

The first application of machine learning was to rank our 34 features using the Relief algorithm, which scores features based on their ability to correctly classify data points to their assigned classes. Each feature vector in the data set was labeled first (assigned a class) using the user role label associated with the system that produced the data. The features in the labeled vectors were then ranked. This process was repeated using the operating system type of the source system as the assigned class.

The second machine learning application was to test how well a J48 decision tree could discriminate between the different users or groups, based on a subset of the features. This test was one of our earlier investigations into group network features. As such, it was based on a subset of the features in Table 2 (items 1-3, 7, 9-11, 13, 18-20, 25-27, 29- 31, 33, 34). Another key difference was that the feature interval for these tests was an hour, rather than 15 minutes. As other tests performed on both the hour long intervals and 15 minute intervals (Bin ratio and Kolmogorov-Smirnov similarity measures, k-means clustering) yielded highly similar results, the results of this test is included in this report.

The classifier was applied in three ways. First, each feature vector was labeled using the hardware address of the user system that produced the data, and the labeled vectors were used to train/test the classifier using 10-fold cross validation. We next tested the data samples using group name labels, again using 10-fold cross validation. As an added check, we tested against sets of pseudo-groups, similarly sized groups of users drawn randomly from the unlabeled (”Others”) data set. This was repeated five times with different random groups, and the mean false positive rates, false negative rates and F-scores reported.

The third machine learning approach tested was to apply k-means clustering to the data. Clustering multi-dimensional data is a well understood method of determining whether data sets exhibit well defined groupings of values. To see if the data samples from the same group would cluster in similar patterns (and differently from other groups), we clustered the data with k=5. Feature values were standardized to range in value from -1 to +1.

IV. IV. RESULTS

H. CORRELATION

One view of how features might relate to a user's role or the operating system of the computer used can be obtained by correlating feature values with the categories of interest. We labeled each system features vector with additional fields (0 or 1) representing membership in one of the role based groups. The correlation values for each feature against the user's role were ordered in terms of absolute value, and are presented in Table 3.

Role	Correlation	Feature
Admin	0.3004	bpp
	0.2845	portsPerFlow
	0.2063	addrDist
	-0.1869	serverRatio
	0.1623	addrPerFlow
PhD student	0.2218	port53Bytes
	0.2087	countEmpties
	-0.185	notTcpUdp
	-0.173	direction
	-0.1604	portDist
Professor	-0.1779	ipDist
	0.1601	notTcpUdp
	0.1573	serverRatio
	-0.1544	duration
	0.1453	maxSrcPortIn
Research Assoc	-0.2712	flagMetric
	-0.264	notTcpUdp
	0.25967	duration
	0.2144	secsPP
	0.163	ipDist
Lecturer	0.1673	addrDist
	0.1623	portDist
	0.1307	notTcpUdp
	0.1158	packets
	0.1112	protocol

Table 3: Top Correlated Features for Roles

The top correlated features for the primary operating systems observed on the network are show in Table 4.

OS	Correlation	Feature
XP	0.4783	notTcpUdp
	0.2867	addrDist
	-0.2389	bpp
	0.1933	protocol
	-0.1852	flowInt
Windows 7	0.3884	portDist
	0.2367	addrDist
	0.2001	direction
	0.1751	bpp
	0.1653	portsPerFlow
Mac	-0.2376	notTcpUdp
	0.1978	UDP
	0.1885	duration
	-0.1783	addrDist
	-0.1736	countEmpties
Linux	-0.4294	addrDist
	-0.3576	notTcpUdp
	-0.3516	portDist
	0.2015	portBytes
	0.1779	addrGrams

Table 4: Top Correlated Features for Operating Systems

As can be seen from these tables, the largest correlation value for role based groups is approximately 0.30 (Admin group, bytes per packet feature), while there are two correlation absolute values for operating systems larger than 0.4 in absolute value. These are (XP, notTcpUdp) at approximately 0.48 and (Linux, addrDist) at approximately 0.43.

I. SIMILARITY MEASUREMENT

The ranges in similarity measures for comparing the PhD students to the Professors group are shown in Figures 2-4 for each of the similarity measures. In these graphs, a similarity measure of 1.0 would mean that the distributions were identical, whereas values closer to zero would show no commonality in the distributions. The Feature Index along the X axis refers to the features listed in Table 2.

For the plots in these figures, groups are compared by measuring the similarity values between each member of one group with each member of the other group. If the groups are the same, systems are not compared with themselves. In the charts, PhD Student to PhD student group similarity measures are shown in blue, while the PhD student vs. Professor Group distributions are shown in green.

As can be seen in the graphs the similarity measure ranges for all features overlapped significantly. This pattern was observed for all group vs. group comparisons; i.e. it appeared that for any feature chosen, the ranges of the intragroup similarity measures was consistently on a par with intergroup similarity measure ranges.

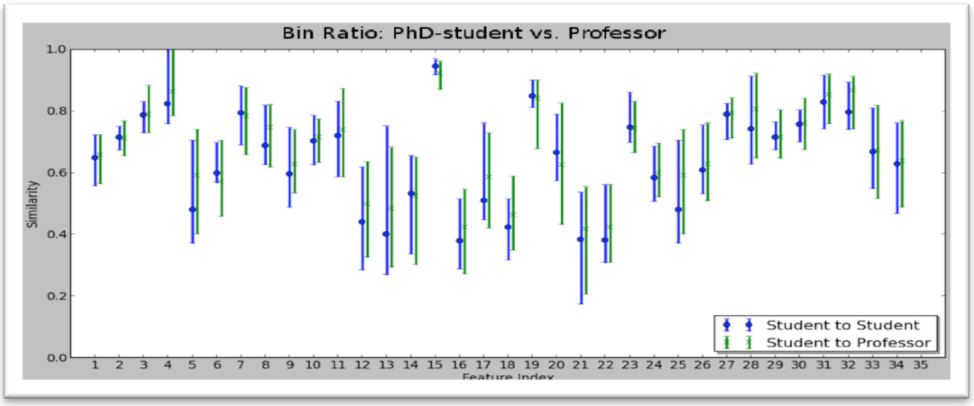

Figure 2: Bin Ratio Measure

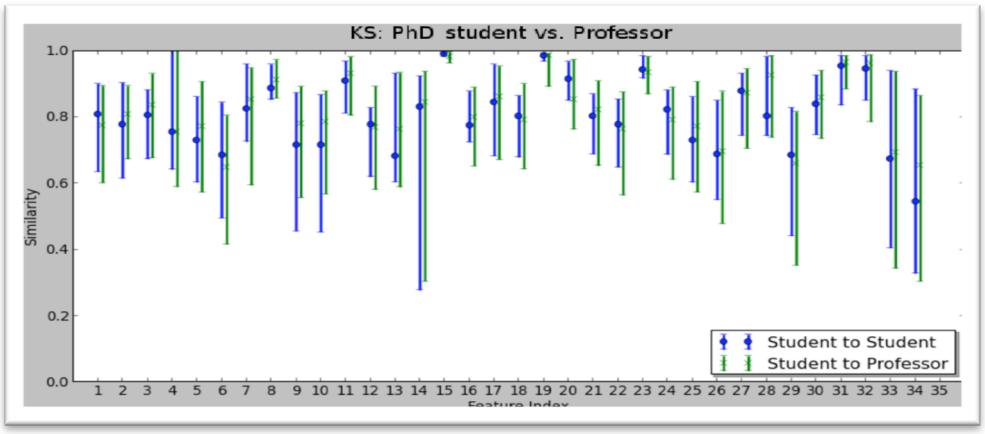

Figure 3: Kolmogorov-Smirnov Measure

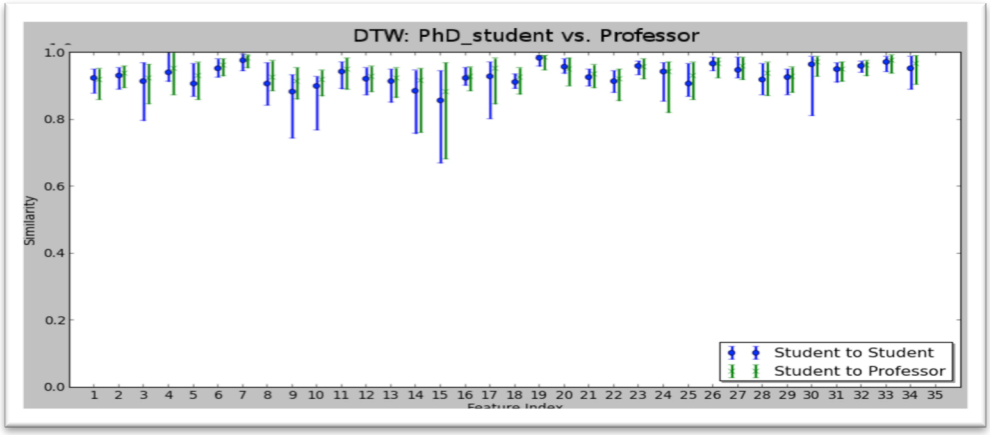

Figure 4: DTW Measure

J. MACHINE LEARNING

1) Relief Algorithm: To evaluate the suitability of our 34 features, we applied the Relief algorithm to rank them using group labels as target classes. The Relief algorithm is instance based (like our data), and provides a relevance weight to each feature based on its ability to predict the given class. The top five ranking features for classifying user roles were addrDist, duration, portDist, protocol, and ipDist, based on Relief ranking. The top five ranking features for operating systems were portDist, addrDist, protocol, ipDist and flagMetric. The Relief scores for the operating system labels were higher, with a range of 0.044-0.098 for the top five versus 0.039-0.052 for the role based group labels. For comparison, the Relief scores for the classic Fisher's iris plant petal dataset [16] (used for data mining training) range from 0.13-0.372.

2) Decision Tree: The results of the J48 decision tree testing are shown in Table 5. This test was run to determine whether the use of groups as a category provides any improvement in the ability of a classifier to identify feature sets derived from each group. In other words, does traffic from a set group have common characteristics that make the data more amenable to classification?

In addition to the role-based and random groups, we also tested the classifier on data labeled by the machine address of the source system. The table reports the average statistics for each group type. Weighted averages are computed for the role-based groups to account for their different sizes.

We note that although the decision tree was able to classify both individual user and groups fairly well using our features subset, the classifier performed only slightly worse (F-scores of 0.69 vs. 0.75 and false positive (FP) rates of 29.2% vs. 24.4%) at classifying the randomly selected groups of users.

Group Type	FP Rate	FN Rate	F-Score	Rule Count
Role-based	24.40%	24.60%	0.751	675
Random	29.20%	29.90%	0.692	769
Individual	26.40%	29.50%	0.716	891

Table 5: Classification Accuracies Using Decision Tree

This result implies that, at least for the features tested, the role based groups did not exhibit strong common characteristics to make them more separable as a group. This is also reflected in the large number of rules required for the classification (approximately 1/8th the number of samples), reflecting fairly low level classifications. Still, the classifier performed better with the role labeled data set and required fewer rules as compared to the system level or pseudo group data sets. While statistically not strong evidence, this implies some level of feature commonality may have enabled greater generalization by the classifier.

These results must be caveated by the fact that the dataset used was based on hour long intervals rather than 15 minute intervals. Based on the close similarity of results of other

tests using both interval periods, we expect that later tests using the full feature set and 15 minute interval based features would result in the same findings.

3) Clustering: After the data samples from the different users were clustered using k-means, we tabulated the percentage of samples assigned to each of the five clusters (Table 6) for each system. Unsurprisingly, we did not find an Administrative cluster, a PhD student cluster and so on. Instead, the data samples from the different systems were allocated in varying proportions across the five clusters. One possible interpretation of this would be that the cluster centers corresponded to common sets of user activities, and the allocations reflected the percent of time spent in these activities.

Group	A	B	C	D	E
Lecturer	79.05	0.48	0.16	4.13	16.19
	73.22	0.50	0.00	0.38	25.91
	58.87	0.39	7.84	1.93	30.98
	2.83	0.17	1.67	8.67	86.67
	8.97	0.85	5.98	23.08	61.11
	21.41	0.32	7.99	2.24	68.05
Admin	31.87	0.29	2.92	2.34	62.57
	3.88	0.38	92.49	0.00	3.25
	41.98	0.65	1.30	8.75	47.33
	25.41	0.50	2.13	1.63	70.34
PhD_student	11.57	0.15	3.24	34.41	50.62
	42.99	0.31	0.62	1.25	54.83
	13.51	0.00	30.81	14.05	41.62
	4.96	0.00	44.33	5.67	45.04
	18.65	0.00	23.78	6.38	51.19
	19.50	0.00	4.15	5.39	70.95
Research_Assoc	94.12	0.00	0.00	0.75	5.13
	9.38	0.52	1.30	10.94	77.86
	66.83	0.00	4.88	0.25	28.04
	2.71	0.29	69.71	5.00	22.29
Professor	62.95	0.38	0.00	6.63	30.04
	21.23	0.00	3.73	1.94	73.10
	57.07	0.38	0.38	2.88	39.30
	28.98	0.38	4.61	12.86	53.17
	14.14	0.50	0.88	0.00	84.48
	61.80	0.00	1.12	0.00	37.08
	66.33	0.50	0.50	13.39	19.27
	6.49	0.70	0.88	22.11	69.82
	80.35	0.50	5.76	0.25	13.14
	7.51	0.00	4.38	1.75	86.36
	19.92	0.57	51.04	0.57	27.89
	0.50	0.00	0.00	8.64	90.86
	9.89	0.00	0.63	6.51	82.98
	54.08	0.47	0.63	4.39	40.44
	92.99	0.38	0.13	0.88	5.63

Table 6: K-means Feature Vector Clusters

Reviewing the tabulated data, it did not immediately appear that these splits of sample data were consistent within the different groups. To reduce the five dimensional vectors

(one per cluster) down to three dimensions, we performed a Principal Components analysis of the tabulated results. Figure 8 shows a scatter plot based on three of the Principal Component axes.

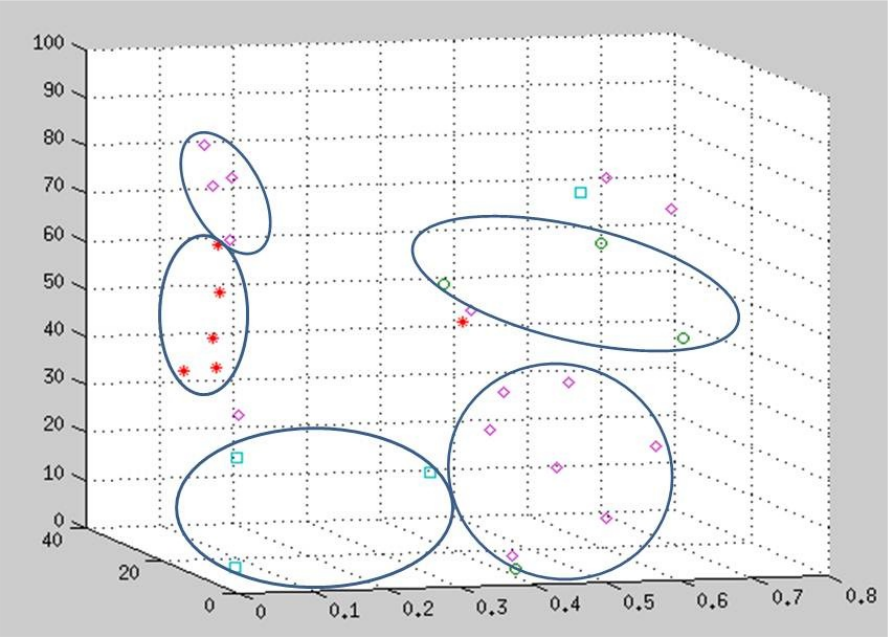

Figure 5: Principal Components View of Clustering Data

While not showing a definitive separation of the five groups, Figure 5 does seem to show some natural clustering for subsets of the groups. If our interpretation of clusters corresponding to common tasks is valid, then this apparent grouping would reflect patterns in the way user groups allocate their time.

V. V. DISCUSSION

The results of our investigations provided a qualified validation of our original hypothesis, i.e. that user roles have an effect on the network behaviors observed.

Correlation: While correlating the feature values with binary indicators of group membership is a fairly simple test, it was informative. None of the features displayed a correlation to any of the role based groups with a value greater than about 0.3. Correlation of the features with the operating system of the user's computer displayed significantly higher values (0.4783 for notTcpUdp to XP, 0.3884 for portDist to Windows 7). Many campus users employ virtual machines on their computers (running different operating systems), which could "muddy the waters" relating to mapping operating systems to network features. Even with this, the relationship between the features studied and host operating systems was notably stronger than that between the features and user roles.

Similarity Measures: The three system similarity measures provide methodologies for comparing systems based on a selected set of network features. The Bin Ratio and Kolmogorov-Smirnov based methods provide new approaches for comparing features based on differences in feature distributions. The Dynamic Time Warping, or feature sequence analysis approach provides a means of evaluating sequences of network features for establishing system similarities. While the system similarity measures investigated showed significant range overlap between intra-group and inter-group settings, this was of course dependent on the group labels used. The labels were an external construct; further investigations into how these features and similarity measures form natural groupings should provide insight into other useful applications.

Machine Learning: The classification tests using machine learning techniques did provide insight into the level of association between the Netflow based features and user role groups. Features scored higher when tested using the Relief algorithm against operating system labels than against user group labels, indicating system configurations have a stronger impact on network traffic. The use of fewer rules by the decision tree and the (slightly) higher F-scores for the group labeled data appeared to indicate the existence of group commonalities, but with only a weak effect on system classification. K-means clustering of the data with the number of clusters equal to the number of roles did not initially show a correlation between cluster centers and the selected groups. This is not in itself surprising, as there are any number of different reasons some users and systems may appear more similar (common operating systems, software loads, working schedule, interests, etc.). That said, the scatter plot in Figure 5 indicates to us that the group commonalities lay in the manner of how group members split their time at tasks.

Because users are individuals and we rarely see rigidly defined roles in organizations, identifying the impact of user roles on network traffic is inherently difficult. Having ground truth about both users and their computer systems was an essential aspect of enabling our methodology for analyzing network traffic. With it, we not only could

identify, extract and label network traffic passing to/from those systems, but also identify a set of non-volunteer systems to use as a control group. For investigating the null hypothesis, i.e. that Netflow based statistical features are not shaped by the role of the user generating them. Having a control group enables a differential analysis between labeled and unlabeled data sets.

Applying a diverse set of features, covering a range of different network traffic characteristics, was also an enabler for our investigation. It allowed some measure of generalization about how interval based statistical features may (or may not) be applicable in characterizing a specified set of groups. If all of the features perform poorly in establishing normal behavioral ranges for individual groups, it can be taken as an indicator that this class of feature types will not do well for this purpose.

The general consistency of range overlaps when comparing groups, regardless of which of our features were examined, indicates to us that context free Netflow based statistical features provide an incomplete view in defining group behavior. In a diverse network environment such as found on a typical campus, what signal might be there relating features to groups may be obscured by the variations in features provided by different system configurations.

So how to follow up on these results? The feature set examined for this analysis was far from complete [17], so more exploration of potential features is needed. The features tried to date contained no contextual information, i.e. no semantic meaning was attached to specific ports or distant IP addresses.

Temporal context was present and used to compute similarity using Dynamic Time Warping, but there are many more ways temporal patterns can be scrutinized. Investigating temporal aspects such as using frequency based analyses, identifying flow sequences with tools such as Hidden Markov Models, classifying system based on n-gram sequences of distant IP addresses, and many other approaches could bear fruit in this area. Varying the interval time might lead to different results, although prior experiments using hour long intervals yielded very similar outcomes. Finally, playing with the Netflow generation parameters to keep flows short for greater temporal resolution may enable more definitive results.

Semantic context can be added by assigning meanings to sets of distant IP addresses and ports used by the systems. Some groups may favor sets of web servers relevant to their group roles. Specific services (reflected by ports visited) may be more necessary to a given group.

Of course there is always the null hypothesis, i.e. that people are individuals and behave as such within their assigned roles. The J48 decision tree classified the individual systems based on the data set fairly well. This ability to recognize individual systems probably enabled it to recognize the randomly generated groups nearly as accurately. In other words, individual traits in traffic activities might dominate the group effect, if present. This might explain the range overlaps between intra-group and inter-group similarity

measures (Figures 2-4). The role group effect was obscured by other, stronger influences on network traffic patterns.

One underlying assumption not mentioned to this point is the stability of the distance metrics over time. To assess this we performed pairwise similarity computations between each of the four work weeks of data for each user, using the Bin Ratio and Kolmogorov-Smirnov derived methods. In other words, we evaluated how each user's traffic is self-similar week to week. Both methods generated consistently high similarity scores, appearing to better support the intuition that user feature distributions should persist over reasonable time intervals. This was especially true of the Kolmogorov-Smirnov distance measure (averaging about 0.9 for most features).

VI. VI. CONCLUSIONS

While a challenging endeavor, being able to compare user behaviors on the network in a meaningful way is essential. All too often it is the users on the network that are the (insider) threat. Using a set of readily obtained network traffic features, we applied known machine learning techniques to look for indications of role based commonalities in user network behavior. We also created and tested several approaches to comparing the similarities between users and/or groups, to aid in identifying deviations in behavior. In doing so, we found several indications that some group based commonalities exist and that feature sets can be effectively compared. This work only scratches the surface of investigating the impact of roles on computer usage behavior; continued research is needed to achieve the end goal of this research: a practical tool that leverages roles and other similar information to create more robust envelops of normal traffic flows and uses them to reduce false positives.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] D. E. Denning, "An intrusion-detection model," IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, vol. 13, no. 2, pp. 222–232, 1987.

[2] S. Nellikar, "Insider threat simulation and performance analysis of insider detection algorithms with role based models," Master's thesis, University of Illinois at Urbana-Champaign, May 2010.

[3] G. F. Anderson, D. A. Selby, and M. Ramsey, "Insider attack and real-time data mining of user behavior," IBM Journal of Research and Development, vol. 51, no. 3.4, pp. 465 –475, may 2007.

[4] J. Park and J. Giordano, "Role-based profile analysis for scalable and accurate insider-anomaly detection," in Performance, Computing, and Communications Conference, 2006. IPCCC 2006. 25th IEEE International, April 2006, pp. 7 pp.–470.

[5] S. Nellikar, D. M. Nicol, and J. J. Choi, "Role-based differentiation for insider detection algorithms," in Proceedings of the 2010 ACM workshop on Insider threats, ser. Insider Threats '10. New York, NY, USA: ACM, 2010, pp. 55–62. [Online]. Available: http://doi.acm.org/10.1145/1866886.1866897

[6] V. Frias-Martinez, "Behavior-based admission and access control for network security," Ph.D. dissertation, Columbia University, 2008.

[7] J. McHugh, R. McLeod, and V. Nagaonkar, "Passive network forensics: behavioral classification of network hosts based on connection patterns," SIGOPS Oper. Syst. Rev., vol. 42, pp. 99–111, April 2008. [Online]. Available: http://doi.acm.org/10.1145/1368506.1368520

[8] T. Furlong, "Tools, data, and flow attributes for understanding network traffic without payload," Master's thesis, Carlton University, April 2007.

[9] D. Rossi and S. Valenti, "Fine-grained traffic classification with NetFlow data," in Proceedings of the 6th International Wireless Communications and Mobile Computing Conference, ser. IWCMC '10. New York, NY, USA: ACM, 2010, pp. 479–483. [Online]. Available: http://doi.acm.org/10.1145/1815396.1815507

[10] T. Karagiannis, K. Papagiannaki, N. Taft, and M. Faloutsos, "Profiling the end host," in Proceedings of the 8th international conference on Passive and active network measurement, ser. PAM'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 186–196. [Online]. Available: http://portal.acm.org/citation.cfm?id=1762888.1762913

[11] F. Giroire, J. Chandrashekar, G. Iannaccone, K. Papagiannaki, E. M. Schooler, and N. Taft, "The cubicle vs. the coffee shop: behavioral modes in enterprise end-users," in Proceedings of the 9th international conference on Passive and active network measurement, ser. PAM'08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 202–211. [Online]. Available: http://portal.acm.org/citation.cfm?id=1791949.1791977

[12] S. Coull, F. Monrose, and M. Bailey, "On Measuring the Similarity of Network Hosts: Pitfalls, New Metrics, and Empirical Analyses," Proceedings of the 18th Annual Network & Distributed System Security Symposium, Feb. 2011.

[13] D. . C. J. Berndt, "Using Dynamic Time Warping to Find Patterns in Time Series," Workshop on Knowledge Discovery in Databases, vol. AAAI-94, pp. 229–248, 1994.

[14] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and

I. H. Witte, “The weka data mining software: An update,” SIGKDD Explorations, vol. 11, no. 1, 2009.
[15] T. Curk, J. Demsar, Q. Xu, G. Leban, U. P. I. Bratko, G. Shaulsky, and B. Zupan, “Microarray data mining with visual programming,” Bioinformatics, vol. 21, pp. 396–398, Feb. 2005. [Online]. Available: http://bioinformatics.oxfordjournals.org/content/21/3/396.full.pdf
[16] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” Annals of Human Genetics, vol. 7, no. 2, pp. 179–188, 1936. [Online]. Available: http://dx.doi.org/10.1111/j.1469-1809.1936.tb02137.x
[17] D. Andrew Moore, “Discriminators for use in flow-based classification,” Department of Computer Science, Queen Mary, University of London, Mile End Road, London E1 4NS, UK, Tech. Rep., updated August 2005.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Geoffrey Xie
   Naval Postgraduate School
   Monterey, CA 93943

2. Robert Beverly
   Naval Postgraduate School
   Monterey, California

3. Neil Rowe
   Naval Postgraduate School
   Monterey, CA 93943